Math Clinic Proposal Draft

Komi Agbo, Dalton Burke, Nick Mako, James Vance

September 30th, 2019

1 Technical Description

1.1 Purpose

Sam's Hauling, Inc. provides small dumpsters of various sizes to homeowners, contractors, realtors and property managers throughout the metro Denver area. They have a limited number of trucks with which to do these pickups and deliveries and in addition some customers set time windows for said pickups and deliveries. In addition, they have multiple depots that they store the dumpsters and the trucks have varying capacities. Currently, the pickup and delivery of these dumpsters is scheduled by a single individual using only Microsoft Excel. That means this problem is at the very least a Multiple Depot Vehicle Routing Problem. Their method for solving it thus far is very time intensive and unlikely to give an optimal solution or even a solution that would pass against a reasonable heuristic. It is quite likely then, that the implementation of some heuristic or metaheuristic could increase the efficiency of this process by a significant margin.

The basic goal of ours is to find a feasible solution that minimizes the expected routing costs while satisfying the customer demands and the constraints related to the number of vehicles and vehicle capacities. In order to do so we intend to build a model which will give a reasonably good solution to this MDVRP and pass against a reasonable heuristic saving Sam's Hauling a significant amount of time.

1.2 Goals and Needs

The basic goal here is to find a feasible solution that minimizes the expected routing costs while satisfying the customer demands, and the constraints related to the number of vehicles, and the vehicles capacity.

1.3 Project Objectives and Requirements

The overarching objective of our group is to create a solution concept for the problem presented by Sam's hauling. During this project it will be necessary for our group to become proficient in using LaTeX to prepare documents and writing code using Python. In order to accomplish this, we will first need to formally define the problem. To do so we will have to determine all variables and constraints of interest and write them out both mathematically and then programmatically within Python. Once we have a formal definition of the problem we will need to determine a method of solution.

1.3.1 Solution

The method of solution can be looked at in a number of parts:

Part 1: Resolving edge cases. In order to simplify the problem a method to deal with the cases where the smallest and largest trucks are needed can be dealt with first. The method to do this will form the first step in the initial solution.

Part 2: The initial solution. Once the edge cases are resolved a method to fully determine the initial solution will need to be defined based on the definition of the problem and the constraints.

Part 3: The Metaheuristic. Once a method for determining an initial solution is decided a metaheuristic must be decided on based on the requirements determined when defining the problem

Part 4: Heuristics. Once a metaheuristic is chosen the heuristics that will be used should be determined if necessary.

1.3.2 Implementation

Once all the parts of the solution have been determined the implementation will need to be created. This will require the writing of code in Python

1.3.3 Testing

Once the implementation has been written we will need to test our solution using real data provided by Sam's hauling. The purpose of this is to show to ourselves and to Sam's how well our solution works in practice. Testing using exaggerated datasets to show how effective the solution would be scaled up would also need to be done so that Sam's can be confident that the solution will work for any reasonable level of future growth.

1.3.4 Finishing

Once we have created and tested our solution, we will need to prepare a report describing everything we have done in a professional manner that explains to Sam's Hauling our process, our solution, and its value to them.

1.4 Measuring Success and Communicating Results

We will compare network flow model against reasonable heuristic arguments. We want to get successful: this means that the cost of realizing our project will be less than the cost we previously proposed as prospective cost.

We will do comparisons between the different steps of the activities. In case the cost gets higher, we are not successful and will revise our model to solve the issue.

Our final goal is to show our audience that our model is efficient and presents routes with least costs. This report will have parts that will explain the efficiency of the model we have built.

We will send our presentation to the company representatives for their appreciations. We believe that our audience will get persuaded that our model is convincing with least costs and efficient route schedules. All these ideas will get proved by the distances between batteries and the each-step time spent.

The functions used with relative variables will be explained. It will get clear to everyone in the audience that we got an efficient proposal with reduced or least cost.

2 Management Plan

Our group has agreed to meet on Mondays at 10:00am, and potentially Wednesday mornings as needed. Here is a rough breakdown of our project:

1. Learn Python, get development environment setup and integrated with git.
2. Draft initial architecture, how components (inputs, outputs, UI, etc.) will interact.
3. Refine model, choose heuristics and methods.
4. Write code to run simplified configurations, ignoring most constraints at first.
5. Create simple visualizations.
6. Add on constraints one at a time, working out errors/bugs along the way.
7. Test real data, debugging
8. Present software to client